Amendments to the Specification:

Please replace paragraph [0001] with the following amended paragraph:

This application claims priority to U.S. Provisional Application Serial No. 60/400,391 titled "JSM Protection," filed July 31, 2002, incorporated herein by reference. This application also claims priority to EPO Application No. 03291927.6, filed July 30, 2003 and entitled "A Multi-Processor Computing System Having A Java Stack Machine And A RISC-Based Processor," incorporated herein by reference. This application also may contain subject matter that may relate to the following commonly assigned co-pending applications incorporated herein by reference: "System And Method To Automatically Stack And Unstack Java Local Variables," Serial No. 10/632,228, filed July 31, 2003, Attorney Docket No. TI-35422 (1962-05401); "Memory Management Of Local Variables," Serial No. 10/632,067, filed July 31, 2003, Attorney Docket No. TI-35423 (1962-05402); "Memory Management Of Local Variables Upon A Change Of Context," Serial No. 10/632,076, filed July 31, 2003, Attorney Docket No. TI-35424 (1962-05403); "A Processor With A Split Stack," Serial No. 10/632,079, filed July 31, 2003, Attorney Docket No. TI-35425(1962-05404); "Using IMPDEP2 For System Commands Related To Java Accelerator Hardware," Serial No. 10/632,069, filed July 31, 2003, Attorney Docket No. TI-35426 (1962-05405); "Test With Immediate And Skip Processor Instruction," Serial No. 10/632,214, filed July 31, 2003, Attorney Docket No. TI-35427 (1962-05406); "Test And Skip Processor Instruction Having At Least One Register Operand," Serial No. 10/632,084, filed July 31, 2003, Attorney Docket No. TI-35248 (1962-05407); "Synchronizing Stack Storage," Serial No. 10/631,422, filed July 31, 2003, Attorney Docket No. TI-35429 (1962-05408); "Methods And Apparatuses For Managing Memory," Serial No. 10/631,252, filed July 31, 2003, Attorney Docket No. TI-35430 (1962-05409); "Write Back Policy For Memory," Serial No. 10/631,185, filed July 31, 2003, Attorney Docket No. TI-35431 (1962-05410); "Methods And Apparatuses For Managing Memory," Serial No. 10/631,205, filed July 31, 2003, Attorney Docket No. TI-35432 (1962-05411); "Mixed Stack-Based RISC Processor," Serial No. 10/631,308, filed July 31, 2003,

Attorney Docket No. TI-35433 (1962-05412); "Processor That Accommodates Multiple Instruction Sets And Multiple Decode Modes," Serial No. 10/631,246, filed July 31, 2003, Attorney Docket No. TI-35434 (1962-05413); "System To Dispatch Several Instructions On Available Hardware Resources," Serial No. 10/631,585, filed July 31, 2003, Attorney Docket No. TI-35444 (1962-05414); "Micro-Sequence Execution In A Processor," Serial No. 10/632,216, filed July 31, 2003, Attorney Docket No. TI-35445 (1962-05415); "Program Counter Adjustment Based On The Detection Of An Instruction Prefix," Serial No. 10/632,222, filed July 31, 2003, Attorney Docket No. TI-35452 (1962-05416); "Reformat Logic To Translate Between A Virtual Address And A Compressed Physical Address," Serial No. 10/632,215, filed July 31, 2003, Attorney Docket No. TI-35460 (1962-05417); "Synchronization Of Processor States," Serial No. 10/632,024, filed July 31, 2003, Attorney Docket No. TI-35461 (1962-05418); "Conditional Garbage Based On Monitoring To Improve Real Time Performance," Serial No. 10/631,195, filed July 31, 2003, Attorney Docket No. TI-35485 (1962-05419); "Inter-Processor Control," Serial No. 10/631,120, filed July 31, 2003, Attorney Docket No. TI-35486 (1962-05420); "Cache Coherency In A Multi-Processor System," Serial No. 10/632,229, filed July 31, 2003, Attorney Docket No. TI-35637 (1962-05421); and "Concurrent Task Execution In A Multi-Processor, Single Operating System Environment," Serial No. 10/632,077, filed July 31, 2003, Attorney Docket No. TI-35638 (1962-05422).

Please replace paragraph [0010] with the following amended paragraph:

Figure 4 shows various registers used in the JSM of Figures 1 and 3;~~and~~

Please replace paragraph [0011] with the following amended paragraph:

Figure 5 shows an exemplary format of a test and skip instruction in accordance with the preferred embodiment of the invention; and[[.]]

Please add the following paragraph after paragraph [0011]:

Figure 6 shows a method in accordance with embodiments of the invention.

Please replace paragraph [0027] with the following amended paragraph:

The test with immediate and skip instruction 228~~238~~ in Figure 5 generally operates in one of at least two modes depending on the state of at least one bit in register reference Rd 232. The Rd reference in Figure 5 preferably includes four bits (Rd[3:0]) with Rd[3] being the most significant bit and occupying bit location 232a. The modes preferably are determined by the most significant bit Rd[3]. In some embodiments one mode is specified when Rd[3] is a logic "0", while another mode is specified when Rd[3] is a "1". In the former mode, Rd 232 references one of the registers from R0-R7, while in the latter mode, Rd references R8-R15. In other embodiments, a logic "1" for Rd [3] may specify the former mode and a logic "0" for Rd [3] may specify the latter mode, or a bit besides Rd [3] could be used to specify the mode.

Please replace paragraph [0028] with the following amended paragraph:

In accordance with the preferred embodiment, as noted above when Rd [3] is a 0, the register Rd 232 encodes one of a first group of registers, preferably registers R0-R7 in Figure 4. If the immediate value V does not match the contents of the eight lower bits of the register referenced by Rd, then the instruction that follows the test and skip instruction 228~~238~~ is "skipped[[.]]" or is not executed. Skipping the next instruction means that the subsequent instruction, which may have already been fetched by fetch logic 154, is not permitted to complete through the processor's pipeline. Skipping the subsequent instruction may occur by replacing the instruction with a "no operation" (NOP) instruction which is permitted to complete but, by its nature, generally does nothing. Skipping the subsequent instruction may be performed other ways as well, such as by flushing the subsequent from the processor's pipeline.

Please replace paragraph [0028] with the following amended paragraph:

If Rd [3] is a logic 1, register reference Rd 232 encodes one of a second group of registers, preferably registers R8-R15. In this mode, the processor 102 generally examines the state of one or more bits, but not necessarily all bits, in register Rd. As such, the immediate value V is used to mask the contents of register Rd 232. This may be accomplished in the preferred embodiment by logically AND'ing register Rd with the immediate value V on a bit-by-bit basis. The mask (immediate value V) may comprise 0's in the bit places that are not of interest, and 1's in the bit places that are of interest. Once the contents of register Rd 232 are masked, the bits corresponding to the bit positions of the mask having a 1 are examined to determine whether they are a 0 or a 1. If the contents of register Rd, after masking, are not all 0's, then the next instruction following the test and skip instruction 228~~238~~ is skipped as described above. If, however, the masked register Rd is all 0's, the subsequent instruction is executed and not skipped.

Please add the following paragraph after paragraph [0033]:

Figure 6 shows a method 600 in accordance with embodiments of the invention. As shown in Figure 6, the method 600 comprises executing a test and skip instruction that includes an immediate value and a reference to a register (block 602). The method 600 then comprises performing a comparison using the immediate value and a register value stored in the referenced register (block 604). Finally, the method 600 comprises executing or skipping a subsequent instruction based on the comparison (block 606).